

What is claimed is:

- 1 1. A system for pre-compiling a source cursor into a target library
2 cache, comprising:
3 at least one source cursor stored in a source library cache, each source
4 cursor comprising a statement with a shareable part and a non-shareable part;
5 an extraction process selectively copying the source cursor by extracting
6 the shareable part of the statement from the source library cache; and
7 a compilation process pre-compiling the shareable part of the extracted
8 source cursor into a target cursor without execution.
- 1 2. A system according to Claim 1, further comprising:
2 a lookup function creating a hash value from a text statement
3 corresponding to the extracted source cursor, comparing the hash value to a set of
4 target cursors stored in the target library cache and retrieving a reference pointer
5 upon locating a matching target cursor.
- 1 3. A system according to Claim 2, further comprising:
2 a build function requesting a context area upon failing to locate a matching
3 target cursor, loading the requested context area and building a new target cursor
4 in the target library cache.
- 1 4. A system according to Claim 1, further comprising:
2 a parent cursor storing the target cursor as a parsed representation of a text
3 statement corresponding to the extracted source cursor, the parent cursor
4 comprising at least one child cursor.
- 1 5. A system according to Claim 4, further comprising:
2 for each session, the compilation process creating at least one child cursor
3 for each text statement having identical text and different objects.
- 1 6. A system according to Claim 4, further comprising:
2 for each session, the compilation process creating at least one child cursor
3 for each text statement having different session environments.

1 7. A system according to Claim 1, further comprising:
2 a target node asynchronously warming the target library cache prior to a
3 switchover.

1 8. A system according to Claim 1, further comprising:
2 a target node asynchronously warming the target library cache prior to an
3 unplanned failover.

9. A system according to Claim 1, wherein the extraction process extracts data selected from the group consisting of at least one of statement text, statement type, parsing user and parsing schema; parsing session environment; parsed representation and execution plan; and bind variable data.

1 10. A system according to Claim 1, wherein the extracted statement is
2 written in a structured database language comprising at least one of SQL and
3 PL/SQL.

1 11. A method for pre-compiling a source cursor into a target library
2 cache, comprising:
3 storing at least one source cursor in a source library cache, each source
4 cursor comprising a statement with a shareable part and a non-shareable part;
5 selectively copying the source cursor by extracting the shareable part of
6 the statement from the source library cache; and
7 pre-compiling the shareable part of the extracted source cursor into a
8 target cursor without execution.

1 12. A method according to Claim 11, further comprising:
2 creating a hash value from a text statement corresponding to the extracted
3 source cursor;
4 comparing the hash value to a set of target cursors stored in the target
5 library cache; and
6 retrieving a reference pointer upon locating a matching target cursor.

1 13. A method according to Claim 12, further comprising:
2 requesting a context area upon failing to locate a matching target cursor;
3 loading the requested context area; and
4 building a new target cursor in the target library cache.

1 14. A method according to Claim 11, further comprising:
2 storing the target cursor as a parsed representation of a text statement
3 corresponding to the extracted source cursor, the target cursor comprising a parent
4 cursor and at least one child cursor.

1 15. A method according to Claim 14, further comprising:
2 for each session, creating at least one child cursor for each text statement
3 having identical text and different objects.

1 16. A method according to Claim 14, further comprising:
2 for each session, creating at least one child cursor for each text statement
3 having different session environments.

1 17. A method according to Claim 11, further comprising:
2 asynchronously warming the target library cache prior to a switchover.

1 18. A method according to Claim 11, further comprising:
2 asynchronously warming the target library cache prior to an unplanned
3 failover.

1 19. A method according to Claim 11, further comprising:
2 extracting data selected from the group consisting of at least one of
3 statement text, statement type, parsing user and parsing schema; parsing session
4 environment; parsed representation and execution plan; and bind variable data.

1 20. A method according to Claim 11, wherein the extracted statement
2 is written in a structured database language comprising at least one of SQL and
3 PL/SQL.

1 21. A computer-readable storage medium holding code for performing
2 the method according to Claim 11.

1 22. A system for staging a pre-compiled cursor in a warmed instance
2 cache, comprising:

3 a hash value created from a source cursor extracted from a source library
4 cache, the source cursor comprising a shareable part and a non-shareable part;
5 a compilation process comparing the hash value to one or more target
6 cursors maintained in a target library cache and retrieving a reference pointer to
7 an address of a matching target cursor.

1 23. A system according to Claim 22, further comprising:
2 an open function opening a cursor definition entry in the target library
3 cache.

1 24. A system according to Claim 23, further comprising:
2 a parse function instantiating the target cursor into the target library cache.

1 25. A system according to Claim 24, further comprising:
2 a bind function binding each input variable in the shareable part of the
3 target cursor.

1 26. A system according to Claim 25, further comprising:
2 a describe function describing type definitions for each input variable in
3 the target cursor without execution.

1 27. A system according to Claim 26, further comprising:
2 a close function closing the target cursor.

1 28. A method for staging a pre-compiled cursor in a warmed instance
2 cache, comprising:
3 creating a hash value from a source cursor extracted from a source library
4 cache, the source cursor comprising a shareable part and a non-shareable part;

5 comparing the hash value to one or more target cursors maintained in a
6 target library cache; and
7 retrieving a reference pointer to an address of a matching target cursor.

1 29. A method according to Claim 28, further comprising:
2 opening a cursor definition entry in the target library cache.

1 30. A method according to Claim 29, further comprising:
2 instantiating the target cursor into the target library cache.

1 31. A method according to Claim 30, further comprising:
2 binding each input variable in the shareable part of the target cursor.

1 32. A method according to Claim 31, further comprising:
2 describing type definitions for each input variable in the target cursor
3 without execution.

1 33. A method according to Claim 32, further comprising:
2 closing the target cursor.

1 34. A computer-readable storage medium holding code for performing
2 the method according to Claim 28.

1 35. A method, comprising:
2 executing a database statement in a first database instance;
3 sending the database statement from the first database instance to a second
4 database instance;
5 in the second database instance, generating and storing a structure
6 required to prepare the database statement for execution in the second database
7 instance;
8 receiving from a user or application a request to execute the database
9 statement in the second database instance; and
10 after receiving the request, using the structure to execute the database
11 statement in the second database instance.

1 36. A method according to Claim 35, wherein the structure is a parse
2 tree for the database statement.

1 37. A method according to Claim 35, wherein the structure is an
2 execution plan for the database statement

1 38. A method according to Claim 35, the sending operation occurs in
2 anticipation of a planned shutdown of the first database instance.

1 39. A method according to Claim 35, wherein the sending operation
2 occurs in anticipation of an unplanned shutdown of the first database instance that
3 may possibly occur in the future.

1 40. A method, comprising:
2 receiving a database statement for execution in a first database instance;
3 generating in the first database instance, a structure required to prepare the
4 database statement for execution;
5 executing the first database statement in the first database instance;
6 sending the structure from the first database instance to a second database
7 instance;
8 receiving from a user or application a request to execute the database
9 statement in the second database instance; and
10 after receiving the request, using the structure to execute the database
11 statement in the second database instance.

1 41. A method according to Claim 40, wherein the structure is a parse
2 tree for the database statement.

1 42. A method according to Claim 40, wherein the structure is an
2 execution plan for the database statement.

1 43. A method according to Claim 40, the sending operation occurs in
2 anticipation of a planned shutdown of the first database instance.

